



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

| APPLICATION NO. | FILING DATE | FIRST NAMED INVENTOR | ATTORNEY DOCKET NO. | CONFIRMATION NO. |
|--|-------------|----------------------|---------------------|------------------|
| 10/820,897 | 04/08/2004 | Maxim A. Ivanov | M1103.70546US00 | 8868 |
| 45840 7590 02/16/2010 WOLF GREENFIELD (Microsoft Corporation) C/O WOLF, GREENFIELD & SACKS, P.C. 600 ATLANTIC AVENUE BOSTON, MA 02210-2206 | | | | |
| EXAMINER KIM, PAUL | | | | |
| ART UNIT | | PAPER NUMBER | | |
| 2169 | | | | |
| MAIL DATE | | DELIVERY MODE | | |
| 02/16/2010 | | PAPER | | |

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary

Application No.

10/820,897

Applicant(s)

IVANOV, MAXIM A.

Examiner

PAUL KIM

Art Unit

2169

Period for Reply -- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 23 December 2009.
- 2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1,3-10,12-16 and 31 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1,3-10,12-16 and 31 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
 2. ☐ Certified copies of the priority documents have been received in Application No. _____.
 3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- 1) ☐ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3) ☐ Information Disclosure Statement(s) (PTO/SB-06)
Paper No(s)/Mail Date _____

- 4) ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date _____
- 5) ☐ Notice of Informal Patent Application
- 6) ☐ Other: _____

DETAILED ACTION

1. This Office action is responsive to the following communication: Amendment filed on 23 December 2009.
2. Claims 1, 3-10, 12-16, and 31 are pending and present for examination. Claims 1 and 9 are in independent form.

Response to Amendment

3. No claims have been amended.
4. No claims have been cancelled.
5. No claim has been newly added.

Claim Rejections - 35 USC § 103

6. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

7. **Claims 1, 3-10, 12-16, and 31** are rejected under 35 U.S.C. 103(a) as being unpatentable over Golden (U.S. Patent No. 6,925,631), filed on 8 December 2000, in view of Vosburgh (U.S. Patent No. 7,089,533), filed on 1 August 2001, published on 14 October 2004, and issued on 8 August 2006, in further view of Murthy et al (U.S. Patent No. 7,096,224, hereinafter referred to as MURTHY), filed on 27 September 2002, and issued on 22 August 2006, and in further view of Wong et al, US Patent No. 7,092,950 (hereinafter referred to as WONG), filed on 29 June 2001 and issued on 15 August 2006.
8. **As per independent claims 1 and 31**, Golden, in combination with VOSBURGH, MURTHY, and WONG, discloses:

Art Unit: 2169

A method for facilitating parsing XML data, the method comprising:

receiving a plurality of user-defined parsing functions {See Golden, C4:L6-17, wherein this reads over "the invocation of software components mapped to the tags is performed during the parsing process"}, the parsing functions being members of a user class library, the functions each containing a custom parsing code written by a user to customize the parsing of the content of XML elements {See Golden, C4:L39-43, wherein this reads over "the discrete software components are classes of an object-oriented programming language, like Java" and "[t]hey can likewise be procedures or functions of a procedural programming language"};

receiving a parsing map for mapping each of a plurality of XML elements to one of the user-defined parsing functions {See Golden, C4:L26-38, wherein this reads over "[f]or tags which are mapped, in the disclosed embodiments, the mapping is a single-valued function"}, the parsing map describing each XML element by an XML element name and identifying the class member associated with the XML element {See Vosburgh, C4:L44-60, wherein this reads over "[t]he XML session manager 102 maps XML files to an object model 110 of the application 111 and vice versa using mapping meta-data that defines how XML files map to the object model"; and C5:L11-23, wherein this reads over "the mapping meta-data may defined the referenced class and the XML elements used to define the relationship"} to be used as a callback method for parsing the content of the XML element;

creating a parser to pre-parse XML source data {See MURTHY, C25:L55-C26:L20, wherein this reads over "[i]f the document is XML, a pre-parse step is performed, where enough of the resource is read to determine the XML schemaLocation and namespace of the root element in the document"}, the parser including a parsing agent, the parsing agent automatically generating a parsing state machine based on the XML element names defined in the parsing map {See WONG, C13:L1-42, wherein this reads over "in addition to the keywords and templates that were discussed, GDL also includes a syntax which allows the appropriate representation and interpretation of entries" and "a GDL Parser is a state-machine, the states of the Parsing engine correspond to Parsing Contexts"};

exposing the mapping to the parser via a communication channel {See Golden, C5:L2-10, wherein this reads over "the discrete software components comprises a method, which is invoked as the start-tag of a tag, is parsed (called 'init method') and a further method, which is invoked as the end-tag of a tag, is parsed"};

receiving an event for the XML element from an event-based reader of XML data containing the element {See Golden, C4:L10-16, wherein this reads over "[t]he preferred parser (an event-driven Application Programming Interface (API) is the 'simple API for XML' ('SAX') parser"; and C5:L11-22, wherein this reads over "the 'behavior' that is induced by a certain tag is not fixed, but can be changed depending on the context in which the input stream is parsed, or any other internal or external conditions"};

pre-parsing the content of the XML element from the XML source data using the parsing state machine {See Golden, C4:L35-38, wherein this reads over "there is a one-to-one mapping between all mapped tags and software components. In other words, to each mapped tag corresponds exactly one software component"}, the pre-parsing comprising identifying the user-defined parsing function to which the XML element is mapped by the parsing map; and

sending the pre-parsed content of the XML element via the communication channel to the user-defined parsing function {See Golden, C5:L44-62, wherein this reads over "[a]s a tag of the input stream is parsed, the software component mapped to it is invoked" and

"[a]nother possible source for XML input is a database, since data can advantageously be stored in databases in the form of XML documents or document fragments. Such an XML database output forms an input for the disclosed embodiments, and as a tag is parse, a software components mapped to it is invoked").

While Golden may fail to expressly disclose the identification of a class member associated with the XML element, Vosburgh discloses an XML session manager that maps XML elements to a referenced class. Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the above invention suggested by Golden with the invention as disclosed by Vosburgh.

One of ordinary skill in the art would have been motivated to do this modification so that the class member may be used in executing the parser functions which parse the content of the XML element.

Additionally, while Golden may fail to expressly disclose the method step of creating a parse to pre-parse XML source data, Murthy discloses a pre-parse step that is performed on an XML document. Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the above invention suggested by Golden with the invention as disclosed by Murthy.

Additionally, while the combination of Golden, Vosburgh and Marthy may fail to disclose that said method step include the generation of a parsing state machine, Wong discloses a method wherein a parsing state machine receives Templates may contain Arbitrary Value context in XML. See Wong, column 16, lines 44-61. Furthermore, Wong discloses that said contexts are used in that "a GDL Parser is a state-machine, the state of the Parsing engine correspond to Parsing Contexts." Accordingly, it would have been obvious to one of ordinary skill in the art that the combination of the aforementioned prior art references would result in the modified invention of defining a parsing function using mapped XML elements found within a parsing map, and wherein the parsing function would contain a parsing state machine which was generated in accordance with the XML elements found within said parsing map.

Art Unit: 2169

One of ordinary skill in the art would have been motivated to do this modification so that a pre-parser may be utilized in the parsing of an XML source data and said parse data mapped accordingly to a received event.

9. **As per dependent claim 3**, Golden, in combination with VOSBURGH, MURTHY, and WONG, discloses:

The method of claim 2, wherein sending the pre-parsed content of the mapped XML element via the communication channel to the parsing function includes looking up the class member identified as being associated with the XML element {See Vosburgh, C5:L11-23, wherein this reads over "the mapping meta-data may defined the referenced class and the XML elements used to define the relationship"}, and sending the pre-parsed content of the XML element to the associated class member {See Vosburgh, C7:L18-36}.

While Golden may fail to expressly disclose the identification of a class member associated with the XML element, Vosburgh discloses an XML session manager that maps XML elements to a referenced class. Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the above invention suggested by Golden with the invention as disclosed by Vosburgh.

One of ordinary skill in the art would have been motivated to do this modification so that the class member may be used in executing the parser functions which parse the content of the XML element.

10. **As per dependent claim 4**, Golden, in combination with VOSBURGH, MURTHY, and WONG, discloses:

The method of claim 2, wherein the parsing function is a reusable object to which the XML element has been previously mapped, and mapping the XML element to the parsing function includes:

creating the parsing map describing the XML element and identifying the reusable object associated with the XML element {See Vosburgh, C5:L11-23, wherein this reads over "the mapping meta-data may define the referenced class and the XML elements used to define the relationship"}; and

joining the reusable object to the other parsing functions described in the parsing map {See Vosburgh, C5:L11-23, wherein this reads over "[f]or one-to-one and one-to-many relationships, meta-data indicates which elements in this (source) class's XML document are used to construct a 'key' to the other (target) class's XML document"}.

Art Unit: 2169

11. **As per dependent claims 5 and 13**, Golden, in combination with VOSBURGH, MURTHY, and WONG, discloses:

The method of claim 1, wherein the agent is an implementation class member and the communication channel is an interface to the implementation class member that enables the mapping to be exposed to the agent automatically (See Vosburgh, C6:L13-15, wherein this reads over "[t]he runtime API 103 is an interface to link the client application 111 and the XML session manager 102. The runtime API 103 defines the XML session object").

While Greenfield may fail to expressly disclose the identification of a class member associated with the XML element, Vosburgh discloses an XML session manager (i.e. an interface to the implementation class member) that maps XML elements to a referenced class. Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the above invention suggested by Golden with the invention as disclosed by Vosburgh. Additionally, it is noted that the present claim recites the intended use of "enabl[ing] the mapping to be exposed to the agent automatically" such that the aforementioned limitation is not afforded patentable weight.

One of ordinary skill in the art would have been motivated to do this modification so that the class member may be used in executing the parser functions which parse the content of the XML element.

12. **As per dependent claim 6**, Golden, in combination with VOSBURGH, MURTHY, and WONG, discloses:

The method of claim 1, wherein the event-based reader of XML data is a SAX reader, and receiving the event for the mapped XML element includes selecting from a plurality of events that have been pushed by the SAX reader only those events that are associated with the mapped XML element (See Golden, C8:L56-C9:L61, wherein this reads over "[t]he SAX parser, an event-driven API, is used for the parsing process. The application registers an event handler to a parser object that implements the org.sax.Parser interface. The event handler interface DocumentHandler is called whenever an element is found in the input stream").

13. **As per dependent claim 7**, Golden, in combination with VOSBURGH, MURTHY, and WONG, discloses:

The method of claim 1, wherein pre-parsing the content of the XML element includes at least one of verifying a structure of the XML element relative to other XML elements occurring in the XML data, verifying a consistency of the XML element, extracting an attribute of the XML element, and collecting a content of the XML element (See Golden, C5:L44-62, wherein this reads over "[a]s a tag of the input stream is parsed, the software component mapped to it is invoked, for example, a software component for accessing a database. Information

Art Unit: 2169

retrieved from the database may then be built into a document page"; and "as a tag is parse, a software component mapped to it is invoked").

14. **As per dependent claim 8**, Golden, in combination with VOSBURGH, MURTHY, and WONG, discloses:

The method of claim 1, further comprising:

mapping an XML element that was previously mapped to an existing parsing function {See Golden, C4:L26-38, wherein this reads over "[f]or tags which are mapped, in the disclosed embodiments, the mapping is a single-valued function"};

joining the existing parsing function to the created parsing function {See Golden, C6:L51-67};

sending the pre-parsed content of the mapped XML element via the communication channel to the joined parsing functions {See Golden, C5:L44-62, wherein this reads over "[a]s a tag of the input stream is parsed, the software component mapped to it is invoked" and "[a]nother possible source for XML input is a database, since data can advantageously be stored in databases in the form of XML documents or document fragments. Such an XML database output forms an input for the disclosed embodiments, and as a tag is parse, a software components mapped to it is invoked"}.

15. **As per independent claim 9**, Golden, in combination with VOSBURGH, MURTHY, and WONG, discloses:

A system for parsing XML data, the system comprising:

a library of custom parsing functions to parse content of XML elements {See Golden, C16:L25-67, wherein this reads over "[t]he XBF engine 13 processes the XML input document 14 as described in the context of FIGS. 5 and 6, using bindings 12 which define the mapping between the tags in the XML input document 14 and classes"};

a parser having a map that associates custom parsing functions with XML elements {See Golden, C4:L6-17, wherein this reads over "the invocation of software components mapped to the tags is performed during the parsing process"};

a communication channel;

an agent that obtains the content of an XML element on behalf of the parser in accordance with the map {See Golden, C5:L2-10, wherein this reads over "the discrete software components comprises a method, which is invoked as the start-tag of a tag, is parsed (called 'init method') and a further method, which is invoked as the end-tag of a tag, is parsed"}, wherein the map is accessed via the communication channel, and further where the agent passes the content to the associated custom parsing function via the communication channel {See Golden, C5:L44-62, wherein this reads over "[a]s a tag of the input stream is parsed, the software component mapped to it is invoked" and "[a]nother possible source for XML input is a database, since data can advantageously be stored in databases in the form of XML documents or document fragments. Such an XML database output forms an input for the disclosed embodiments, and as a tag is parse, a software components mapped to it is invoked"}.

Art Unit: 2169

16. **As per dependent claim 10**, Golden, in combination with VOSBURGH, MURTHY, and WONG, discloses:

The system of claim 9, wherein the library of custom parsing functions is a class library of members that receive content from the agent via the communication channel {See Golden, C4:L39-43, wherein this reads over "the discrete software components are classes of an object-oriented programming language, like Java" and "[t]hey can likewise be procedures or functions of a procedural programming language"}.

17. **As per dependent claim 12**, Golden, in combination with VOSBURGH, MURTHY, and WONG, discloses:

The system of claim 9, wherein at least one of the parsing functions is a reusable object to which an XML element has been previously associated {See Vosburgh, C5:L11-23, wherein this reads over "the mapping meta-data may define the referenced class and the XML elements used to define the relationship"}, and the parser joins the reusable object to the other parsing functions in the map {See Golden, C6:L51-67}.

While Greenfield may fail to expressly disclose that parsing function is a reusable object to which an XML element has been previously associated, Vosburgh discloses that there may be a plurality of XML elements which may be mapped to the same object. Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the above invention suggested by Golden with the invention as disclosed by Vosburgh.

One of ordinary skill in the art would have been motivated to do this modification so that the multiple objects would not have been created wherein one reusable shared object would suffice in providing for the parsing functions.

18. **As per dependent claim 14**, Golden, in combination with VOSBURGH, MURTHY, and WONG, discloses:

The system of claim 9, further comprising an event-based reader, wherein the agent obtains the content of the XML element on behalf of the parser, including handling events generated for the XML element by the event-based reader {See Golden, C5:L2-10, wherein this reads over "the discrete software components comprises a method, which is invoked as the start-tag of a tag, is parsed (called 'init method') and a further method, which is invoked as the end-tag of a tag, is parsed"}.

19. **As per dependent claim 15**, Golden, in combination with VOSBURGH, MURTHY, and WONG, discloses:

Art Unit: 2169

The system of claim 14, wherein the event-based reader of XML data is a SAX reader (See Golden, C4:L10-16, wherein this reads over "[t]he preferred parser (an event-driven Application Programming Interface (API) is the 'simple API for XML' (SAX) parser"; and C5:L11-22, wherein this reads over "the 'behavior' that is induced by a certain tag is not fixed, but can be changed depending on the context in which the input stream is parsed, or any other internal or external conditions"), and handling events generated for the XML element includes at least one of verifying a structure of the XML element relative to other XML elements occurring in the XML data, verifying a consistency of the XML element, extracting an attribute of the XML element, and collecting the content of the XML element (See Golden, C5:L44-62, wherein this reads over "[a]s a tag of the input stream is parsed, the software component mapped to it is invoked, for example, a software component for accessing a database. Information retrieved from the database may then be built into a document page"; and "as a tag is parse, a software component mapped to it is invoked").

20. **As per dependent claim 16**, Golden, in combination with VOSBURGH, MURTHY, and WONG, discloses:

The system of claim 9, wherein the agent obtains the content of the XML element on behalf of the parser using a state machine generated in accordance with the map (See Golden, C4:L35-38, wherein this reads over "there is a one-to-one mapping between all mapped tags and software components. In other words, to each mapped tag corresponds exactly one software component").

Response to Arguments

21. Applicant's arguments filed 23 December 2009 have been fully considered but they are not persuasive.

a. **Claim Rejection under 35 U.S.C. 103**

Applicant asserts the argument that Wong fails to disclose "the parsing agent automatically generating a parsing state machine based on the XML element names defined in the parsing map." See Amendment, page 8. The Examiner respectfully disagrees. Specifically, Applicant asserts that "Wong's state machine is used to parse GDL and is incompatible with XML, as XML violates the syntax." See Amendment, page 8. The Examiner notes that Wong discloses a system wherein "the Arbitrary Value context allows the Parser to handle values that would otherwise violate the GDL syntax." See Wong, col. 16, lines 55-56. Furthermore, Wong discloses that the Arbitrary Value may include other languages like XML. See Wong, col. 16, lines 57-58. Accordingly, wherein XML may violate the GDL syntax, it is noted that Wong discloses that the

Art Unit: 2169

Arbitrary Value context allows the Parser to handle the XML language. Wherein Wong discloses that "a GDL Parser is a state-machine, the states of the Parsing engine correspond to Parsing Contexts" such as an Arbitrary Value Context which may include XML language, it would have been obvious to one of ordinary skill in the art that said disclosure by Wong would read upon the limitation of "the parsing agent automatically generating a parsing state machine based on the XML element names defined in the parsing map." See Wong, col. 13, lines 40-42.

Accordingly, for the aforementioned reasons above, the claim rejections under 35 U.S.C. 103 are maintained.

Conclusion

22. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

23. Any inquiry concerning this communication or earlier communications from the examiner should be directed to PAUL KIM whose telephone number is (571)272-2737. The examiner can normally be reached on M-F, 9am - 5pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tony Mahmoudi can be reached on (571) 272-4078. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Art Unit: 2169

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Tony Mahmoudi/
Supervisory Patent Examiner, Art Unit 2169

Paul Kim
Examiner, Art Unit 2169
Technology Center 2100

/pk/